



UNITED STATES PATENT AND TRADEMARK OFFICE

[Handwritten signature]
UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/632,157	07/31/2003	Timo J. Salo	5577-259	1907
20792	7590	08/11/2006		EXAMINER
MYERS BIGEL SIBLEY & SAJOVEC				PANTOLIANO JR, RICHARD
PO BOX 37428				
RALEIGH, NC 27627			ART UNIT	PAPER NUMBER
			2194	

DATE MAILED: 08/11/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/632,157	SALO ET AL.	
	Examiner Richard Pantoliano Jr	Art Unit 2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 31 July 2003.
 2a) This action is FINAL. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-22 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-22 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on 31 July 2003 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ . |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date <u>20030731</u> . | 6) <input type="checkbox"/> Other: _____ . |

DETAILED ACTION

1. This is the initial office action for Application# **10/632,157** filed on **31 July 2003**.

Claims 1-22 are currently pending and have been considered below.

Information Disclosure Statement

2. Citation No. 6 on the Information Disclosure Statement filed on 31 July 2003 does not supply a date for the disclosed document. The enclosed document also appears to be incomplete. The document is cited as consisting of 26 pages, but only 13 pages were found to be included with the application. It appears that alternating pages of the disclosed document were not included.

3. Examiner has located a complete version of the disclosed reference dated 25 June 2002. If the copy of said cited reference included by Examiner is not an equivalent version of the document submitted by Applicant, Applicant is requested to submit a new copy of said reference and the publishing date of said reference.

Specification

4. The disclosure is objected to because of the following informalities:

a) Pg 9, Line 24: The phrase "...collection of target **EBJs**..." appears to be a typographical error.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

6. **Claims 6 and 20** are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Both claims recite the limitation of checking "the collection registry by the link object in response to the one-to-many or many-to-many association not being materialized...". Figures 6 and 7, and lines 3-31 on page 10 of the disclosure indicate that collection registry is examined directly for the presence of the collection, instead of trying to materialize said collection before examining the collection registry.

Claim Rejections - 35 USC § 101

7. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 9 and 15 rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 9 includes a module programmed to perform several functions within the system. However, the Detailed Description of the specification states that modules can be implemented entirely as software components. Software is considered functional

descriptive material and functional descriptive material *per se* is considered non-statutory subject matter by the Office.

Claim 15 recites "...the computer program product comprising a computer usable storage medium...". Within the Detailed Description of the disclosure, Applicant specifies "transmission media" as possible computer usable storage media. However, the Office considers only media capable of storing and retaining information encoded with computer executable instructions as statutory subject matter. Since transmission media is only capable of transporting computer executable code, and not storing said code, computer executable code on a transmission medium is considered non-statutory subject matter.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. **Claims 1-4, 9-12, 15-18, 21 and 22** are rejected under 35 U.S.C. 102(b) as being anticipated by BEA Systems ("The Weblogic Server EJB Container and Supported Services", http://e-docs.bea.com/wls/docs70/ejb/EJB_environment.html, p1-27, published 25 June 2002).

Claim 1: BEA Systems discloses the method of maintaining association integrity of Enterprise JavaBeans (EJB) comprising:

- a) obtaining a collection of target EJBs that are associated with a source EJB (*Pg 11, paragraph 7*) ("loading related beans"); and
- b) registering the collection of target EJBs in a collection registry (*Pg 11, paragraph 7 and Pg 12, paragraph 3*) (*The "cache" serves as a dedicated location for the related beans*).

Claim 2: BEA Systems discloses the method of **Claim 1**, wherein the registering comprises storing the EJBs in a collection registry upon passivation of the source EJB (*Pg 4, paragraph 1*) (*The disk stores the information for the EJBs that are removed from memory and placed in a disk cache, which serves as another registry, until reactivation*).

- Claim 3:** BEA Systems discloses the method of **Claim 2**, further comprising:
- a) reactivating the source EJB (*Pg 3, Figure 4-2*); and
 - b) fetching the collection of target EJBs that are associated with the source EJB in response to traversing the relationships between the EJBs (*Pg 11, paragraph 7 – pg 12, paragraph 3*) (*Fetching the source EJB invokes the Relationship Caching behavior of the Weblogic Server and causes all of the passivated EJBs to be loaded with the source EJB. The relationships between the source EJB and the associated EJBs is traversed and stored manually by the database developer in the "weblogic-cmp-rdbms-jar.xml" file*).

Claim 4: BEA Systems discloses the method of **Claim 2**, further comprising:

- a) reactivating the source EJB (*Pg 3, Figure 4-2*);
- b) fetching the collection of target EJBs that are associated with the source EJB in response to traversing the relationships between the EJBs (*Pg 11, paragraph 7 – pg 12, paragraph 3*) (*Fetching the source EJB invokes the Relationship Caching behavior of the Weblogic Server and causes all of the passivated EJBs to be loaded with the source EJB. The relationships between the source EJB and the associated EJBs is traversed and stored manually by the database developer in the “weblogic-cmp-rdbms-jar.xml” file*); and
- c) materializing the collection if the source EJB is not registered (*Pg 3, paragraphs 3-6 and Pg 11, paragraph 7 – pg 12, paragraph 3*) (*When a client initially connects to the server, the source EJB is created and loaded into memory. Due to the relationship caching behavior of the server, the loading of the source EJB causes the associated collection of EJBs to be loaded with said source EJB into memory*).

Claim 9: BEA Systems discloses the system for maintaining association integrity of Enterprise JavaBeans (EJBs) comprising:

- a) a collection registry (*Pg 11, paragraph 7 and Pg 12, paragraph 3*) (*The “cache” serves as a dedicated location for the storage of the related beans*);
- b) a module to obtain a collection of EJBs associated with a source EJB and register that collection in the collection registry (*Pg 11, paragraph 7 – pg 12*,

paragraph 3)(The Weblogic Server serves as the software module to accomplish the task of storing the EJBs in the cache. The server loads the associated EJBs and stores them in a cache).

Claim 10: BEA Systems discloses the system of **Claim 9** wherein the associated EJBs are stored in a registry upon passivation of the source EJB (*Pg 4, paragraph 1*) (*The disk stores the information for the EJBs that are removed from memory and placed in a disk cache, which serves as another registry, until reactivation*).

Claim 11: BEA Systems discloses the system of **Claim 10** wherein the module fetches the collection of EJBs upon reactivation of the source EJB (*Pg 11, paragraph 7 – pg 12, paragraph*) (*Fetching the source EJB invokes the relationship caching behavior of the Weblogic Server and causes all of the associated passivated EJBs to be loaded with the source EJB. The relationships between the source EJB and the associated EJBs is traversed and stored manually by the database developer in the “weblogic-cmp-rdbms-jar.xml” file*).

Claim 12: BEA Systems discloses the system of **Claim 10** wherein the module is programmed to:

a) fetch the collection of target EJBs that are associated with the source EJB in response to traversing the relationships between the EJBs (*Pg 3, Figure 4-2 and Pg 11, paragraph 7 – pg 12, paragraph 3*) (*Reactivating the source EJB invokes*

the relationship caching behavior of the Weblogic Server and causes all of the associated passivated EJBs to be loaded from the cache with the source EJB. The relationships between the source EJB and the associated EJBs is traversed and stored manually by the database developer in the “weblogic-cmp-rdbms-jar.xml” file); and

b) materializing the collection if the source EJB is not registered (*Pg 3, paragraphs 3-6 and Pg 11, paragraph 7 – pg 12, paragraph 3*)*(When a client initially connects to the server, the source EJB is created and loaded into memory. Due to the relationship caching behavior of the server, the loading of the source EJB causes the associated collection of EJBs to be loaded with said source EJB into memory).*

Claim 15: BEA Systems discloses the computer-readable medium containing executable instructions for maintaining the association integrity of Enterprise JavaBeans (EJBs), the instructions comprising:

a) instructions for obtaining a collection of target EJBs that are associated with a source EJB (*Pg 11, paragraph 7*)*(“loading related beans”); and*
b) instructions for registering the collection of target EJBs in a collection registry (*Pg 11, paragraph 7 and Pg 12, paragraph 3*)*(The cache serves as a dedicated location for the related beans).*

Claim 16: BEA Systems discloses the computer-readable medium containing executable instructions of **Claim 15** wherein the instructions comprise the step of

storing the associated EJBs in a collection registry upon passivation (*Pg 4, paragraph 1*) (*Pg 4, paragraph 1*) (*The disk stores the information for the EJBs that are removed from memory and placed in a disk cache, which serves as another registry, until reactivation*).

Claim 17: BEA Systems discloses the computer-readable medium containing executable instructions of **Claim 16**, wherein the instructions comprise the step of fetching associated EJBs upon reactivation of the source EJB (*Pg 11, paragraph 7 – pg 12, paragraph*) (*Fetching the source EJB invokes the relationship caching behavior of the Weblogic Server and causes all of the associated passivated EJBs to be loaded with the source EJB. The relationships between the source EJB and the associated EJBs is traversed and stored manually by the database developer in the “weblogic-cmp-rdbms-jar.xml” file*).

Claim 18: BEA Systems discloses the computer-readable medium containing executable instructions of **Claim 16**, wherein the instructions comprise the steps of:
a) fetching associated EJBs upon reactivation of the source EJB (*Pg 11, paragraph 7 – pg 12, paragraph 3*) (*Fetching the source EJB invokes the Relationship Caching behavior of the Weblogic Server and causes all of the passivated EJBs to be loaded with the source EJB. The relationships between the source EJB and the associated EJBs is traversed and stored manually by the database developer in the “weblogic-cmp-rdbms-jar.xml” file*); or

b) materializing the collection if the source EJB is not registered (*Pg 3, paragraphs 3-6 and Pg 11, paragraph 7 – pg 12, paragraph 3*)*(When a client initially connects to the server, the source EJB is created and loaded into memory. Due to the relationship caching behavior of the server, the loading of the source EJB causes the associated collection of EJBs to be loaded with said source EJB into memory).*

Claim 21: BEA Systems discloses the system for maintaining association integrity of Enterprise JavaBeans (EJBs) comprising:

a) means for obtaining a collection of target EJBs that are associated with a source EJB (*Pg 11, paragraph 7*)*("loading related beans"); and*
b) means for registering the collection of target EJBs in a collection registry (*Pg 11, paragraph 7 and Pg 12, paragraph 3*)*(The "cache" serves as a dedicated location for the related beans).*

Claim 22: BEA Systems discloses the system of **Claim 21**, further comprising:

a) means for reactivating the source EJB (*Pg 3, Figure 4-2*);
b) means for fetching the collection of target EJBs that are associated with the source EJB in response to traversing the relationships between the EJBs (*Pg 11, paragraph 7 – pg 12, paragraph 3*)*(Fetching the source EJB invokes the Relationship Caching behavior of the Weblogic Server and causes all of the passivated EJBs to be loaded with the source EJB. The relationships between*

the source EJB and the associated EJBs is traversed and stored manually by the database developer in the “weblogic-cmp-rdbms-jar.xml” file); and

c) means for materializing the collection if the source EJB is not registered (Pg 3, paragraphs 3-6 and Pg 11, paragraph 7 – pg 12, paragraph 3)(*When a client initially connects to the server, the source EJB is created and loaded into memory. Due to the relationship caching behavior of the server, the loading of the source EJB causes the associated collection of EJBs to be loaded with said source EJB into memory*).

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

11. **Claims 5-8, 14,15, 19, and 20** are rejected under 35 U.S.C. 103(a) as being unpatentable over BEA Systems (“The Weblogic Server EJB Container and Supported Services”, http://e-docs.bea.com/wls/docs70/ejb/EJB_environment.html, p1-27, published 25 June 2002) in view of Sun (Enterprise JavaBeans™ Specification Version 2.0”, Sun Microsystems Inc., 22 August 2001).

Claim 5: BEA Systems discloses the method of **Claim 1**, but does not disclose the use of a “link factory” in maintaining the relationships between EJBs in the registry. Sun discloses that, in designing an application using EJBs, the developer is responsible

for specifying the relationships between EJBs (*Sun, pg 129, paragraph 1*) and that said relationships must be maintained using standard application-programming interfaces (*Pg 129-130, "The entity bean provider's programming contract" and pg 134-135, "Semantics of assignment for relationships*). The logic contained within the software module to maintain the relationships and produce that collection of associated EJBs is irrelevant, as long as it provides the basic interfaces required by the EJB specification to communicate with an EJB and produce the specified collection of associated EJBs. Therefore, it would have been obvious to one of ordinary skill in the art to use any means for maintaining the relationships between the EJBs within the system. One would have been motivated by the inherent requirements of the Enterprise JavaBean specification to use any means of maintaining the necessary relationships between the EJBs while using a standard interface to maintain those relationships.

Claim 6: BEA Systems discloses the method of **Claim 2**:

- a) wherein relationships between the source EJB and the collection of EJBs is maintained (*Pg 11, paragraph 7 – pg 12, paragraph*);
- b) wherein the registering comprises:
 - i) creating a collection registry to store the collection of associated EJBs (*Pg 11, paragraph 7 – pg 12, paragraph 3*);
 - ii) managing the collections using the registry (*Pg 11, paragraph 7 – pg 12, paragraph 3*);
- c) wherein fetching comprises:

- i) checking the collection registry to determine if the collection needs to be fetched or materialized (*it is inherent that the WebLogic Server would examine its cache to determine if it needs to return the EJBs from its cache or materialize new versions of those EJBs*);
- ii) returning the related collection of EJBs if the collection was found in the registry (*Pg 11, paragraph 7 – pg 12, paragraph*); and
- iii) materializing the related collection of associated EJBs if the collection was not found in the registry (*Pg 3, paragraphs 3-6 and Pg 11, paragraph 7 – pg 12, paragraph 3*)*(When a client initially connects to the server, the source EJB is created and loaded into memory. Due to the relationship caching behavior of the server, the loading of the source EJB causes the associated collection of EJBs to be loaded with said source EJB into memory). Relationship Caching behavior will cause the other beans to be loaded).*

BEA Systems does not disclose the use of a “link factory” in all of the steps to analyze the relationships maintained between the EJBs and producing a collection of associated EJBs based on that link factory. Sun discloses that, in designing an application using EJBs, the developer is responsible for specifying the relationships between EJBs (Sun, pg 129, paragraph 1) and that said relationships must be maintained using standard application-programming interfaces (*Pg 129-130, “The entity bean provider’s programming contract” and pg 134-135, “Semantics of assignment for relationships*). The logic contained within the software module to maintain the relationships and

produce that collection of associated EJBs is irrelevant, as long as it provides the basic interfaces required by the EJB specification to communicate with an EJB and produce the specified collection of associated EJBs. Therefore, it would have been obvious to one of ordinary skill in the art to use any means for maintaining the relationships between the EJBs within the system. One would have been motivated by the inherent requirements of the Enterprise JavaBean specification to use any means of maintaining the necessary relationships between the EJBs while using a standard interface to maintain those relationships.

Claim 7: BEA Systems and Sun disclose the system of **Claim 6**, but does not disclose the use of a “link factory” to determine what EJBs need to be materialized by examining the relationship defined between the EJBs. Sun discloses that, in designing an application using EJBs, the developer is responsible for specifying the relationships between EJBs (Sun, pg 129, paragraph 1) and that said relationships must be maintained using standard application-programming interfaces (Pg 129-130, “*The entity bean provider’s programming contract*” and pg 134-135, “*Semantics of assignment for relationships*”). The logic contained within the software module to maintain the relationships and produce that collection of associated EJBs is irrelevant, as long as it provides the basic interfaces required by the EJB specification to communicate with an EJB and produce the specified collection of associated EJBs. Therefore, it would have been obvious to one of ordinary skill in the art to use any means for maintaining the relationships between the EJBs within the system and making those relationships

available to be analyzed through the standard application-programming interface. One would have been motivated by the inherent requirements of the Enterprise JavaBean specification to use any means of maintaining the necessary relationships between the EJBs while using a standard interface to maintain those relationships.

Claim 8: BEA Systems and Sun disclose the system of **Claim 6**, but does not disclose the step of returning the target collection of EJBs via a “link factory” when the collection is materialized. Sun discloses that, in designing an application using EJBs, the developer is responsible for specifying the relationships between EJBs (Sun, pg 129, paragraph 1) and that said relationships must be maintained using standard application-programming interfaces (Pg 129-130, “*The entity bean provider’s programming contract*” and pg 134-135, “*Semantics of assignment for relationships*”). The logic contained within the software module to maintain the relationships and produce that collection of associated EJBs is irrelevant, as long as it provides the basic interfaces required by the EJB specification to communicate with an EJB and produce the specified collection of associated EJBs. Therefore, it would have been obvious to one of ordinary skill in the art to use any means for maintaining the relationships between the EJBs within the system. One would have been motivated by the inherent requirements of the Enterprise JavaBean specification to use any means of maintaining the necessary relationships between the EJBs while using a standard interface to maintain those relationships.

Claim 13: BEA Systems discloses the system of **Claim 9**, but does not disclose a “link factory” for use in generating and managing the relationships between the requested EJBs. Sun discloses that, in designing an application using EJBs, the developer is responsible for specifying the relationships between EJBs (Sun, pg 129, paragraph 1) and that said relationships must be maintained using standard application-programming interfaces (Pg 129-130, “*The entity bean provider’s programming contract*” and pg 134-135, “*Semantics of assignment for relationships*”). The logic contained within the software module to maintain the relationships and produce that collection of associated EJBs is irrelevant, as long as it provides the basic interfaces required by the EJB specification to communicate with an EJB and produce the specified collection of associated EJBs. Therefore, it would have been obvious to one of ordinary skill in the art to use any means for maintaining the relationships between the EJBs within the system. One would have been motivated by the inherent requirements of the Enterprise JavaBean specification to use any means of maintaining the necessary relationships between the EJBs while using a standard interface to maintain those relationships.

Claim 14: BEA Systems discloses the system of **Claim 10**, but does not disclose the use of a “link factory” to manage the relationships between the EJBs. Sun discloses that, in designing an application using EJBs, the developer is responsible for specifying the relationships between EJBs (Sun, pg 129, paragraph 1) and that said relationships must be maintained using standard application-programming interfaces (Pg 129-130,

"The entity bean provider's programming contract" and pg 134-135, "Semantics of assignment for relationships). The logic contained within the software module to maintain the relationships and produce that collection of associated EJBs is irrelevant, as long as it provides the basic interfaces required by the EJB specification to communicate with an EJB and produce the specified collection of associated EJBs. Therefore, it would have been obvious to one of ordinary skill in the art to use any means for maintaining the relationships between the EJBs within the system. One would have been motivated by the inherent requirements of the Enterprise JavaBean specification to use any means of maintaining the necessary relationships between the EJBs while using a standard interface to maintain those relationships.

Claim 19: Systems discloses the computer-readable medium containing computer executable instructions of **Claim 15**, further comprising instructions for registering the collection of target EJBs in a collection registry (*Pg 11, paragraph 7 and Pg 12, paragraph 3*) (*The "cache" serves as a dedicated location for the related beans*).

However, BEA Systems does not disclose instructions to use a "link factory" in generating the collection of EJBs in response to traversing the relationships between the EJBs. Sun discloses that, in designing an application using EJBs, the developer is responsible for specifying the relationships between EJBs (Sun, pg 129, paragraph 1) and that said relationships must be maintained using standard application-programming interfaces (*Pg 129-130, "The entity bean provider's programming contract" and pg 134-135, "Semantics of assignment for relationships*). The logic contained within the

software module to maintain the relationships and produce that collection of associated EJBs is irrelevant, as long as it provides the basic interfaces required by the EJB specification to communicate with an EJB and produce the specified collection of associated EJBs. Therefore, it would have been obvious to one of ordinary skill in the art to use any means for maintaining the relationships between the EJBs within the system. One would have been motivated by the inherent requirements of the Enterprise JavaBean specification to use any means of maintaining the necessary relationships between the EJBs while using a standard interface to maintain those relationships.

Claim 20: BEA Systems discloses the computer-readable medium containing computer executable instructions of **Claim 16**, comprising:

- a) instructions for maintaining relationships between the source EJB and the collection of associated EJBs (*Pg 11, paragraph 7 – pg 12, paragraph*);
- b) instructions for registering, wherein the instructions comprise:
 - i) instructions for creating a collection registry to store the collection of associated EJBs (*Pg 11, paragraph 7 – pg 12, paragraph 3*);
 - ii) instructions for managing the collections using the registry (*Pg 11, paragraph 7 – pg 12, paragraph 3*);
- c) instructions for fetching, wherein the instructions comprise:
 - i) instructions for checking the collection registry to determine if the collection needs to be fetched or materialized (*it is inherent that the*

WebLogic Server would examine its cache to determine if it needs to return the EJBs from its cache or materialize new versions of those EJBs); ii) instructions for returning the related collection of EJBs if the collection was found in the registry (Pg 11, paragraph 7 – pg 12, paragraph); and iii) instructions for materializing the related collection of associated EJBs if the collection was not found in the registry (Pg 3, paragraphs 3-6 and Pg 11, paragraph 7 – pg 12, paragraph 3)(When a client initially connects to the server, the source EJB is created and loaded into memory. Due to the relationship caching behavior of the server, the loading of the source EJB causes the associated collection of EJBs to be loaded with said source EJB into memory). Relationship Caching behavior will cause the other beans to be loaded).

BEA Systems does not disclose instructions for a “link factory” analyze the relationships maintained between the EJBs and producing a collection of associated EJBs based on that link factory. Sun discloses that, in designing an application using EJBs, the developer is responsible for specifying the relationships between EJBs (Sun, pg 129, paragraph 1) and that said relationships must be maintained using standard application-programming interfaces (Pg 129-130, “The entity bean provider’s programming contract” and pg 134-135, “Semantics of assignment for relationships). The logic contained within the software module to maintain the relationships and produce that collection of associated EJBs is irrelevant, as long as it provides the basic interfaces required by the EJB specification to communicate with an EJB and produce the

specified collection of associated EJBs. Therefore, it would have been obvious to one of ordinary skill in the art to use any means for maintaining the relationships between the EJBs within the system. One would have been motivated by the inherent requirements of the Enterprise JavaBean specification to use any means of maintaining the necessary relationships between the EJBs while using a standard interface to maintain those relationships.

Conclusion

12. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure:

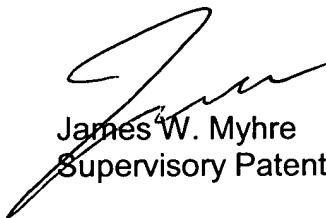
- a) Sharma et al. (*US Pat: 6,877,111*) discloses a system and method for maintaining state of Enterprise JavaBeans;
- b) Clement et al. (*US PG Pub: 2004/0078782*) discloses a system and method for managing workload between Enterprise JavaBeans in a server; and
- c) Messinger et al. (*US Pat: 6,886,041*) discloses a system and method for maintaining multiple dispatch pools to handle incoming requests to an Enterprise JavaBean.

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Richard Pantoliano Jr whose telephone number is (571) 270-1049. The examiner can normally be reached on Monday-Thursday, 8am - 4 pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, James W. Myhre can be reached on (571)270-1065. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

RP
RP
8/03/06



James W. Myhre
Supervisory Patent Examiner